1 Part 2

1.1 Description of the problem

In real applications, the Yelp database is expected to be visited by different groups of people, including customers (users), data analyst (special users), and developers. In this project, this is further divided into five categories:

1. A casual user who uses the application to browse search results. These users do not need to have an account; hence, they cannot submit reviews.

2. Critiques that use the application to browse results just like the casual user, but they also leave reviews for places they visit. A logged in user should only be provided enough privileges to write the review.

3. Business analysts can use the application to produce sales reports and may want to do special data mining and analysis. They cannot perform IUD (Insert/Update/Delete) operations on the database but should have access to creating extra views on the database schema.

4. Developers working with this database are able to create new tables and perform data cleaning and indexing. They are allowed to perform IUD operations on the database.

5. The database admin who has full access over the database.

The principle of granting privilege is to guarantee that each group of people have sufficient permission in order to protect the database. First, the list of all privileges in MySQL 5.7 are listed in Table 0.1, from which we can choose levels for each user group[1].

1.2 Group 1

For the first group of users, they only browse information about the business, including their opening hours, stars, reviews, without signing in so they do not need to write information into the database. In some cases, if the app allows some specific types of anonymous communications, such as marking a review as "cool" or "useful" by a visitor, then the permission should be extended to allow for modification of the count of these tags. However, in this project we assume that the user are not allowed to perform any operations except exploring. Hereby we only grant `SELECT` privilege to the first group of user, which we call `user1`:

```
DROP USER IF EXISTS 'user1'@'%';
CREATE USER user1;
GRANT SELECT ON yelp_db.* TO 'user1'@'%';
```

[1] https://dev.mysql.com/doc/refman/5.7/en/grant.html

Privilege	Meaning and Grantable Levels
ALL [PRIVILEGES]	Grant all privileges at specified access level except GRANT OPTION and PROXY.
ALTER	Enable use of ALTER TABLE. Levels: Global, database, table.
ALTER ROUTINE	Enable stored routines to be altered or dropped. Levels: Global, database, procedure.
CREATE	Enable database and table creation. Levels: Global, database, table.
CREATE ROUTINE	Enable stored routine creation. Levels: Global, database.
CREATE TABLESPACE	Enable tablespaces and log file groups to be created, altered, or dropped. Level: Global.
CREATE USER	Enable use of CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES. Level: Global.
CREATE VIEW	Enable views to be created or altered. Levels: Global, database, table.
DELETE	Enable use of DELETE. Level: Global, database, table.
DROP	Enable databases, tables, and views to be dropped. Levels: Global, database, table.
EVENT	Enable use of events for the Event Scheduler. Levels: Global, database.
EXECUTE	Enable the user to execute stored routines. Levels: Global, database, table.
FILE	Enable the user to cause the server to read or write files. Level: Global.
GRANT OPTION	Enable privileges to be granted to or removed from other accounts. Levels: Global, database, table, procedure, proxy.
INDEX	Enable indexes to be created or dropped. Levels: Global, database, table.
INSERT	Enable use of INSERT. Levels: Global, database, table, column.
LOCK TABLES	Enable use of LOCK TABLES on tables for which you have the SELECT privilege. Levels: Global, database.
PROCESS	Enable the user to see all processes with SHOW PROCESSLIST. Level: Global.
PROXY	Enable user proxying. Level: From user to user.
REFERENCES	Enable foreign key creation. Levels: Global, database, table, column.
RELOAD	Enable use of FLUSH operations. Level: Global.
REPLICATION CLIENT	Enable the user to ask where master or slave servers are. Level: Global.
REPLICATION SLAVE	Enable replication slaves to read binary log events from the master. Level: Global.
SELECT	Enable use of SELECT. Levels: Global, database, table, column.
SHOW DATABASES	Enable SHOW DATABASES to show all databases. Level: Global.
SHOW VIEW	Enable use of SHOW CREATE VIEW. Levels: Global, database, table.
SHUTDOWN	Enable use of mysqladmin shutdown. Level: Global.
SUPER	Enable use of other administrative operations such as CHANGE MASTER TO, KILL, PURGE BINARY LOGS, SET GLOBAL, and mysqladmin debug command. Level: Global.
TRIGGER	Enable trigger operations. Levels: Global, database, table.
UPDATE	Enable use of UPDATE. Levels: Global, database, table, column.
USAGE	Synonym for no privileges

Table 0.1: Privileges available in MySQL

1.3 Group 2

For the second type of user, they are different from casual users in that they may leave reviews or tips on a business. They are logged-in users, so they can interact with other reviews or tips. Therefore, they are granted global `SELECT` privilege, `INSERT` on the review and tip table, `UPDATE` on certain columns in the business table, and table-wise `UPDATE` on user table. The SQL query is shown as follows, similarly we call this `user2`:

```
DROP USER IF EXISTS 'user2'@'%';
CREATE USER user2;
GRANT SELECT ON yelp_db.* TO 'user2'@'%';
GRANT INSERT ON yelp_db.review TO 'user2'@'%';
GRANT INSERT ON yelp_db.tip TO 'user2'@'%';
GRANT UPDATE (stars) ON yelp_db.business TO 'user2'@'%';
GRANT UPDATE (review_count) ON yelp_db.business TO 'user2'@'%';
GRANT UPDATE ON yelp_db.user TO 'user2'@'%';
```

1.4 Group 3

Business analysts are special casual users. Here we assume they are not logged in so they are not expected to change any contents in the database. Therefore, we only add some view-related privileges to this group of users besides those granted to group 1:

```
DROP USER IF EXISTS 'user3'@'%';
CREATE USER user3;
GRANT SELECT, CREATE VIEW, SHOW VIEW ON yelp_db.* TO 'user3'@'%';
```

1.5 Group 4

Group 4 corresponds to normal developers. These people are in charge of the visiting, development and maintenance of database. Therefore we grant them full IUD privileges on the whole database. Also, in case they need to perform automated operations, query optimization or concurrency control, we also grant them with view, routine(function, procedure), index and lock permissions. The SQL queries are as follows:

```
DROP USER IF EXISTS 'user4'@'%';
CREATE USER user4;
GRANT ALTER ROUTINE, CREATE ROUTINE, EXECUTE, # routine related
CREATE VIEW, SHOW VIEW, # view related
CREATE, ALTER, INDEX, REFERENCES, # tables, indexes and keys
DELETE, DROP, INSERT, SELECT, UPDATE # basic operations including IUD
ON yelp_db.* TO 'user4'@'%';
```

1.6 Group 5

Group 5 is the database administrator, so its privilege is all but `GRANT` and `PROXY` options, which should only be done using the root user. In practical use only these two operations

should be done using root user in order to prevent abuse or unexpected threats to the database. The SQL queries are as follows:

```
DROP USER IF EXISTS 'user5'@'%';
CREATE USER user5;
GRANT ALL ON yelp_db.* TO 'user5'@'%';
```